



## 1. Какво е eBay?

Всички знаем или сме ползвали този сайт, така че задълбаването в този въпрос не е необходимо. Накратко, eBay освен че е сайт, е и американска интернет компания, която притежава едноименния сайт за онлайн търговия. В него се продават както нови, така и използвани (втора ръка) стоки. Тези стоки се предлагат под формата на търг или директна продажба на установена цена.

## 2. Какво са микроуслугите?

Преди да преминем към архитектурата на eBay, нека изясним какво представляват микроуслугите.

Първоначално всички системи са били имплементирани монолитно, тоест от горе надолу, ред по ред. Често тези системи са нямали истинска структура, нито път е било обръщано внимание на възможността за поддръжка в бъдеще, което е правело тези системи трудни за поддръжка. Това е особено вярно за онези системи, които са достигали голям обем на сорс код и са били използвани от много потребители.

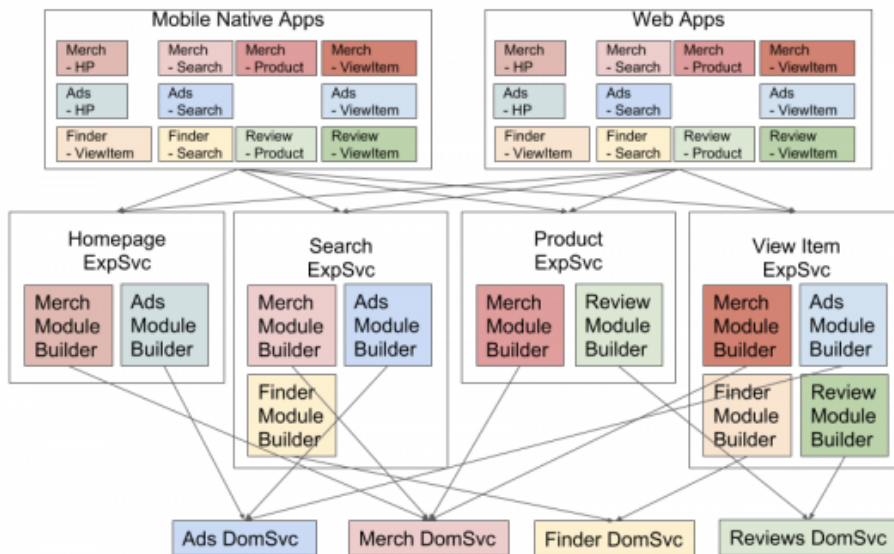
С появата на обектно-ориентираното програмиране се поражда архитектурата, ориентирана към услуги (или така наречената SOA). Това дава началото на създаването на системите като колекция от слабо свързани услуги.

Микроуслугите са следващата еволюция на SOA. При тях, всяка услуга се разработва по отделно, като е възможно всяка да е написана на различен програмен език. Същността на всяка услуга е да изпълнява един use case или бизнес изискване.

## 3. Архитектурата на eBay

Когато през 2011 eBay е интегрирала микроуслугите, компанията е имала 97 милиона активни потребители и 62 милиарда продукти. В един обикновен ден, системите на eBay е трябвало да се справят с огромен трафик - 75 милиарда извиквания към базата данни, 4 милиарда заявки за преглеждане на страници и 250 милиарда заявки за търсене. eBay са знаели, че за да останат конкурентноспособни, те трябвало да предоставят качествени услуги и иновации. Интегрирането на микроуслугите и разделянето на системата на отделни части (бази данни, различни нива/слоеве на приложение, търсачка) е дало възможността на eBay да отговорят на всеувеличаващата се сложност на сорс кода и да подобрят продуктивността на разработчиците. Системата eBay използва над 1000 услуги. За да онагледим как работи една част от системата на eBay, ще покажем следните примери и ще изясним какво е показано върху тях:

Фигура 1



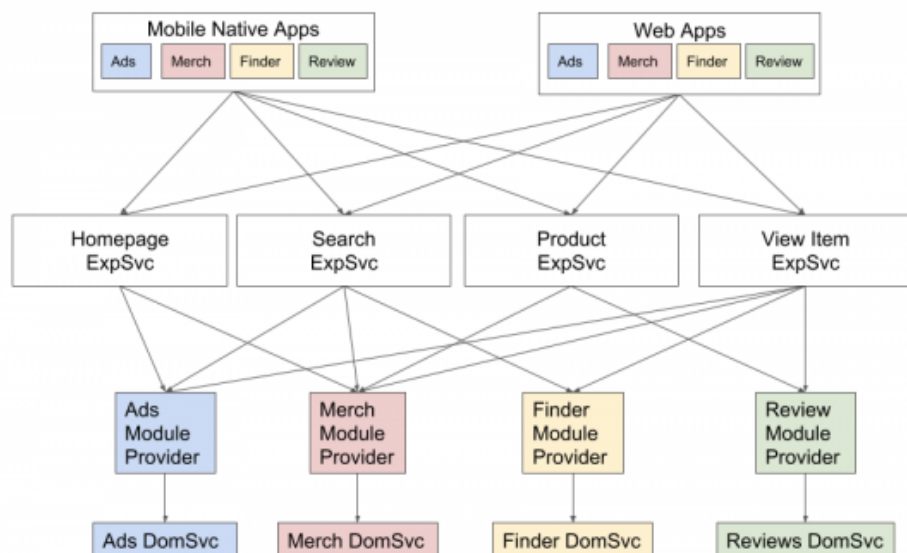
Това е системата на eBay преди интегрирането на микроуслугите. Както виждаме, eBay използва два вида 'experiences' - Mobile Native Apps и Web Apps. Всеки един от тях има своя имплементация на едни и същи компоненти, които обаче се различават по това кой ги извиква - това са всички компоненти, представени на снимката, които са част

и от двата 'experiences'. Тези компоненти извикват едни и същи интерфейси към желаните услуги, като отново всеки интерфейс притежава собствена имплементация на компонент за свързване с услугите. На тази диаграма могат да се видят причините защо eBay са сменили архитектурата на системата:

- С нарастването на броя активни потребители, системата е трябвало да отговаря на повече заявки за търсене, за преглеждане на страници или към базата данни за извличане на информация. Но как е възможно да се разшири системата, когато функционалността ѝ е разделена навсякъде по кода. Ако е трябвало да се реализира нов интерфейс или услуга, то екипа от разработчици, отговарящ за това е трябвало да се свърже с много други екипи, преди да се имплементира. Ето защо времето за добавяне или оправяне на даден компонент/интерфейс/experience/услуга е било много голямо, а както всички знаем времето е и пари все пак.
- Имало е много повторения на код навсякъде в сорс кода. Това е правело системата ненужно сложна и голяма.

Поради тези причини, eBay са си дали сметка, че наистина трябва да подобрят своята система, така че тя да може да се разширява със синхрон с увеличаващите се очаквания на потребителите, както и на техния брой. Резултатът е следната архитектура:

На следващия пример могат да се онагледят всички предимства на употребата на микроуслуги. Може би най-голямата разлика с горната диаграма е това колко по-просто изглежда тази архитектура, отколкото горната. Това се дължи на факта, че тук вместо да бъдат използвани компоненти за една и съща функционалност във всяко едно ниво/слой на абстракция, всъщност отделните функционалности са отделени от останалата част на системата. Сега и experience модулите, и интерфейсите имат



Фигура 2

свои компоненти за комуникация с отделните нива на системата и в тях няма нищо друго, което да се повтаря и да създава излишни сложности. Както виждаме, имаме абстракция на показването на заглавната страница, на търсачката, на информацията за продукт и на показването на продукта. Тези интерфейси от своя страна използват различни видове услуги, за да предоставят желаният отговор, чрез други

интерфейси към тези услуги.

#### 4. Качествени характеристики, които са повлияни от употребата на микрослужите

##### - *Мащабируемост (Scalability)*

Благодарение на употребата на микрослужите, eBay е постигнала много голяма мащабируемост. Това е така, защото отделните функционалности са разработени в рамките на една отделна услуга. Това означава, че когато се забележи някакво натоварване на системата (например в по-натоварени дни като Black Friday), лесно може да се открие натоварения компонент в системата и да се скалира, така че да може да издържи и да отговаря на идващите заявки.

##### - *Възможност за поддръжка (Maintenance)*

Едно от главните предимства на употребата на микрослужите е именно подобрието на възможността за поддръжка. Когато се проектира архитектурата на един софтуер, трябва да се вземат под внимание, неговите бъдещи перспективи - дали ще подлежи на промяна, ако да къде ще бъде тази промяна. Като следствие от това, системата трябва да бъде проектирана по такъв начин, че да позволява лесното интегриране или поправяне на нови модули или компоненти. Микрослужите са вид архитектура, която позволява всичко това да се осъществи по много лесен начин. Нови функционалности се добавят лесно, защото те трябва само и единствено да се свържат с вече съществуващите интерфейси. Също така промяната на една функционалност няма да повлияе изобщо на останалите услуги в системата, поради това, че тя не е свързана с тях. Разбира се същото подлежи и за

интерфейсите и за experience модулите. Това спестява на компанията значителна сума, която иначе би била отделена за поддръжката.

#### - *Изменяемост (Modifiability)*

Това качествено изискване си прилича с горното, защото засяга промяната на системата. Но тук, тази промяна се отнася за промяна по време на разработка. Микроуслугите подобряват изменяемостта на една система, поради същите причини - всяка функционалност е отделена от другите, всеки интерфейс говори само и единствено с услуги и 'experience' модули, а те от своя страна имат връзка само с интерфейси. По този начин, ако даден интерфейс се промени, той евентуално би засегнал само услугите, които е ползвал и 'experience' модулите, с които е общувал. Ако пък бъде добавена нова услуга, то тя няма да засегне други модули в системата - нужно е само свързването ѝ с интерфейси. От тези примери става ясно, че се спестява значително време в промяната на системата, както и пари.

#### - *Преизползваемост (Reusability)*

Тъй като функционалностите са разработени в отделни услуги, които не говорят помежду си, това позволява на компанията да интегрира някоя услуга в бъдещи проекти без да се налагат значителни промени (или понякога изобщо никакви). Ето защо преизползваемостта е подобрена с употребата на микроуслугите.

#### - *Възможност за тестване (Testability)*

Поради отделената функционалност на системата, тестването става много по-лесно ако например е използвана монолитна архитектура. Може да се тестват отделните функционалности и тяхното поведение независимо от конкретната ситуация и по този начин да се открият евентуални грешки, които да бъдат поправени. Въпреки, че тестването в големи система като eBay е много сложен процес, употребата на микроуслугите го смекчават. Възможността за тестването е една важна качествена характеристика, която се цели да открие наличието на грешки, което е особено важно за големите системи, които биват използвани от милиони хора.

## 5. Използвана литература

[Introduction to microservices architecture - June 28, 2018](#)

[10 companies that implemented the microservice architecture and paved the way for others - 02/08/2019](#)

[Microservices at eBay. Part 2: Sharing Modules Across Experience services and Multi-Screen Applications - January 11, 2019](#)

[What eBay looks like under the hood - March 7, 2016](#)